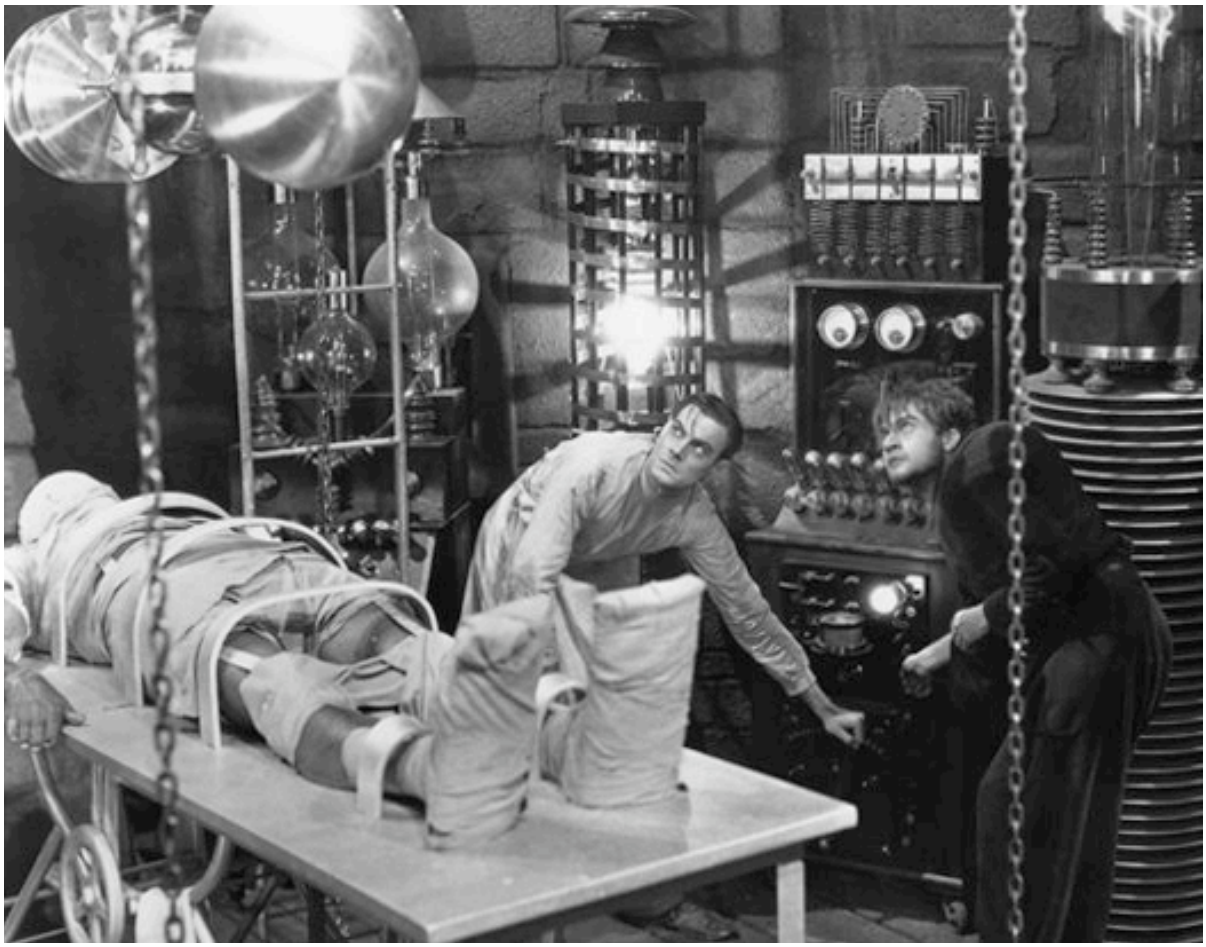
SOFTWARE ENGINEERING PROCESSES AND MANAGEMENT



. . . Stand back Igor! I'm going to start the process!!

WORKSHOP 2
MEET THE PROCESS

INTRODUCTION

The aim of this workshop is to explore the relationships that exist between processes, people, technology and outcomes. This workshop is much more about the logic that you use than the actual answers that you get, as there is often more than one way to choose a process by to achieve an outcome.

Each realisation of such a process has certain advantages and certain costs. The problem for the software engineer is to choose a process that is effective, can be realised with the people and technology available for the project and will deliver the outcomes of the project on time and on budget. In this workshop we ask you to consider one the two case-studies and for your chosen case-study to determine one or two possible processes that may be used to successfully develop each system.

> **NOTE:** We do not expect you to know "the answers". We expect you to think about the question and start putting ideas about process together and discussing the positive and negative aspects of your choices.

The point of this workshop is to get you thinking about the design of processes and to start understanding the *deep* view. For now, however, your task is to design a process that will meet the goals of the language technology project.

But how does one design a process that will deliver a system with a specific set of properties?

SOME GUIDELINES

When designing a process to achieve a goal we usually need to work by making choices in at least three different levels (see Figure 1).

At the *processes* level we often choose a life-cycle process that minimises risk, allows us to build in engineering methods that meet reliability, performance or other specialised requirements and allows us to give good feedback to stakeholders.

At the *phase processes* level we often choose processes that will support the life-cycle in both form and in in philosophy. For example, agile requirements methods often clash with formal processes because they do not provide enough detail for the types of analysis required by formal processes.

Lastly we will often use the term *tool chain* rather than tool set with the idea that the tools and techniques used in one phase can inter-operate with the tools used in other phases. An example is the Rationale suit that allows its requirements tool to inter-operate with Rationale architect.

DESIGNING A PROCESS FOR A PROJECT

Your task in this workshop is simply to design a life-cycle process for the language research platform project. Specifically you are asked to do the following tasks.

Choose a Life-cycle model The first task is to choose a life-cycle model for the language research case study project. When you choose a life-cycle for the project you will need to:

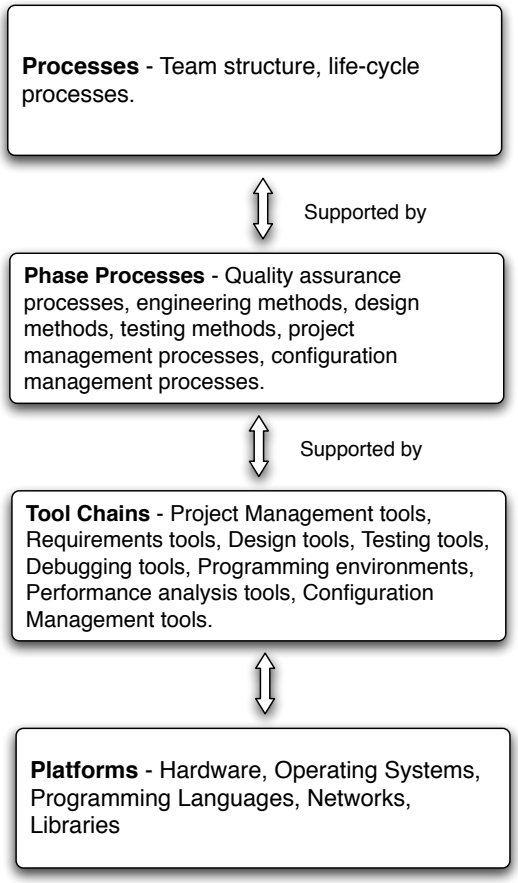


Figure 1: Choices to make when design processes.

- Determine the key characteristics and goals for this project and work out which of the life-cycle models best fits the goals of the project.
- Justify your choice of life-cycle model based on you criteria. There may be no right answer for this — saying that two or even three different approaches fit the criteria and simply choosing one of them is appropriate.

Design a process for each phase of your life-cycle The second task is to use what you know of software engineering ad a good dose of common (or even uncommon) sense try and sketch what process you would put in place to achieve the goals of each phase of your chosen life-cycle process.

Your choices may have an impact on project risk and project life-cycle. What are the impacts of your choices or sub-process have?

How might you achieve a quality software system?

Suggest tools and techniques Again using common sense, uncommon sense and whatever experience you have had, suggest some tools and techniques that you might use to support your processes? How well do the tools and techniques integrate with each other?

Check your process This is a mental exercise but is often valuable when designing processes but list three problems that you foresee in this kind of project — think of late delivery of technology, excessive time slippage, or third-party software that does not meet its specifications — and see how your process would deal with them.